Coupling Tangent-linear and Adjoint Models

Uwe Naumann[1] and Patrick Heimbach[2]

[1] Mathematics and Computer Science Division, Argonne National Laboratory
9700 South Cass Avenue, Argonne, IL 60439, USA
naumann@mcs.anl.gov, http://www.mcs.anl.gov
[2] Earth, Atmospheric and Planetary Sciences, Massachusetts Institute of Technology
heimbach@mit.edu, http://www-eaps.mit.edu

Abstract. We consider the solution of a (generalized) eigenvalue problem arising in physical oceanography that involves the evaluation of both the tangent-linear and adjoint versions of the underlying numerical model. Two different approaches are discussed: First, tangent-linear and adjoint models are generated by the software tool TAF and used separately. Second, the two models are combined into a single coupled derivative model based on optimally preaccumulated local gradients of all scalar assignments.

1 Introduction

A major focus in understanding the ocean circulation's role in the variability, on time scales of decades to millennia and beyond, of the climate system is put on understanding the so-called 'thermohaline' circulation (THC). This refers to the contribution to that aspect of the ocean circulation which is driven by density gradients and thus controlled by temperature and salinity properties and its associated fluxes (but see [1], for problems of its isolated discussion). It is believed to play a crucial role in connecting the surface to the deep ocean through deep-water formation which occur at some isolated convection sites at high latitudes mainly in the North Atlantic, such as the Labrador Sea and the Greenland-Irminger-Norwegian (GIN) Seas.

Because of its relatively small variability over the past few thousand years in contrast to potentially large changes on glacial-interglacial time scales, it's dynamics may be well described by modal behavior of linear dynamics such as stochastically (or realistically) forced damped oscillations. Major changes in the THC are then ascribed to either stochastic forcing across the stability threshold, shifting the system between stable equilibria, or nonlinear effects such as self-sustained oscillations. In addressing what triggers instabilities in the climate system, comparatively little attention has been given to systems which are non-normal, i.e. which exhibit non-orthogonal eigenmodes (beginning with [2]; see [3] and references therein). Such systems, albeit linear, can undergo large transient amplifications. In [4] these ideas were applied to a simplified box model of the THC serves as a paradigm model to capture some essential features of the circulation [5,6]. The authors demonstrate how, for a given optimality condition (norm), a set of initial conditions of temperature and salinity may be determined which maximize the transient amplification of the THC. From an automatic differentiation (AD) [7] point of view, their approach is interesting for two reasons: (i): their calculation involves both the tangent linear (TLM) and the adjoint (ADM) operator of the model, (ii): their approach can be generalized to a coupled atmosphere-ocean model of intermediate complexity, or even to a fully-fledged general circulation model (GCM), for which AD becomes a crucial ingredient in deriving the TLM and ADM.

For the present purpose, our focus remains on the simple box model, for which we wish to derive the TLM and ADM by means of AD. The box model is introduced in the following section. The optimality problem is stated in Section 3. Section 4 briefly describes how the TLM and ADM were generated by means of FastOpt's AD tool TAF (Transformation of Algorithms in Fortran) [8,9]. The main reason for choosing this tool is that it has been applied successfully to generate efficient derivative code of MIT's parallel ocean general circulation model (MITgcm), [10–14], and will thus play an important part in the generalization of the box model study to a GCM.

An algorithm for coupling the TLM and ADM is presented in Section 5. It is based on optimally preaccumulated gradients of scalar assignments [15,16]. Both approaches are compared and conclusions are drawn in Section 7 in the light of current work on a new infra-structure for the implementation of next-generation AD algorithms.

2 A simple model of the thermohaline circulation

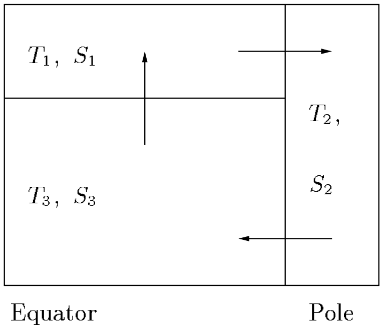

Fig. 1. A simple model of the thermohaline circulation

Some important aspects of linear dynamics of the THC in the North Atlantic are captured by a simple 3-box model [4,17], see Figure 1. Box 1 represents the shallow surface waters between the equator and the sub-polar ocean. Box 2 reflects the dynamics of the polar ocean and its role in deep water formation. Box 3 captures the deep ocean at low and mid latitudes. The model state is represented by properties of temperature T_i and salinity S_i in each of the three boxes, $i = 1, \ldots, 3$. The dynamics is governed by a set of equations; here, we only state the sub-set for box 1 for the case $U \geq 0$, the full set of equations can be found in [17]:

$$\begin{aligned} U &= u_0 \{ \rho_2 - [\delta \rho_1 + (1-\delta)\rho_3] \} \\ \rho_i &= -\alpha T_i + \beta S_i , \qquad i = 1, \ldots, 3 \\ \frac{d}{dt} T_1 &= U (T_3 - T_1)/\mathcal{V}_1 + F_1^{relax} \\ \frac{d}{dt} S_1 &= U (S_3 - S_1)/\mathcal{V}_1 + F_1^{fw} \end{aligned} \tag{1}$$

U denotes the net density-driven transport between the boxes with mean transport u_0, density ρ_i is inferred via a linear equation of state with thermal expansion coefficient α and β, and tracer advection terms dT_1/dt, dS_1/dt for temperature and salinity. $\mathcal{V}_1$ refers to the volume of each box, while F_1^{relax}, F_1^{fw} denote external forcing due to air-sea fluxes of heat (or temperature relaxation) and freshwater, respectively. The full set of parameter choices are given in [4]. The model is stepped forward in time by a standard upwind differencing scheme.

3 The optimality criterion

The time-evolving temperature and salinity in each box together define a six-dimensional state vector

$$\boldsymbol{P} = [T_1, T_2, T_3, S_1, S_2, S_3]^T \quad , \tag{2}$$

which, for the linear system considered here, evolve in time according to

$$\boldsymbol{P}(\tau) = \mathcal{M}(\tau, 0) \boldsymbol{P}(0) \tag{3}$$

with model propagator $\mathcal{M}(\tau, 0)$ between $t = 0$ and $t = \tau$.

Following [4], we may formulate a projection $\boldsymbol{R}$ of the model state onto the THC transport U according to

$$U = \boldsymbol{R} \cdot \boldsymbol{P} \tag{4}$$

and define an associated norm $U^2 = ||\mathbf{X}||^2$ of the THC strength whose matrix representation is of the form

$$X_{ij} = R_i R_j \quad . \tag{5}$$

We then wish to find an initial condition $\boldsymbol{P}(t = 0)$ which maximizes the THC strength

$$J(\tau) = \boldsymbol{P}(\tau)^T \mathbf{X} \boldsymbol{P}(\tau) \tag{6}$$

at some final time $t = \tau$ under the side condition that the initial condition have norm unity for a given (possibly different) norm $\mathbf{Y}$ at $t = 0$,

$$J(0) \; = \; \boldsymbol{P}(0)^T \, \mathbf{Y} \, \boldsymbol{P}(0) \quad . \tag{7}$$

This leads to a constrained optimization problem which can be solved by the method of Lagrange multipliers. The optimal initial conditions are then found to be those, which fulfill the generalized eigenvalue problem

$$M^T(\tau,0)\mathbf{X}\,M\,(\tau,0)\boldsymbol{e} \; = \; \lambda\mathbf{X}\boldsymbol{e} \tag{8}$$

with (generalized) eigenvalues and vectors λ and $\boldsymbol{e}$, respectively. Here, $M(\tau,0)$ denotes the tangent linear operator of the model $\mathcal{M}$, and $M^T(\tau,0)$ its adjoint (transpose).

This problem can be readily solved in MATLAB [4]. However, in the present context of an implementation in FORTRAN, and in view of future extensions to fully-fledged ocean models, an alternative iterative strategy is adopted:

1. At a given iteration n, a given estimate $\boldsymbol{e}_n$ is used to compute

$$\boldsymbol{y}_n \; = \; M^T(\tau,0)\,\mathbf{X}\,M(\tau,0)\boldsymbol{e}_n \tag{9}$$

by means of the tangent linear and adjoint operator

2. $\boldsymbol{y}_n$ and $\boldsymbol{e}_n$ are provided to an Implicitly Restarted Arnoldi Iteration Routine, ARPACK [18], which yields new estimates $\boldsymbol{e}_{n+1}$ and λ_{n+1}.

This method enables an efficient computation of the leading eigenvectors and eigenvalues for large-scale applications, for which the dimension of the state space is large ($\sim 10^7$ for ocean GCM's).

4 Automatic TLM and ADM Generation via TAF

The generation of the tangent-linear model via TAF is straightforward: providing the dependent variables (final state) and independent variables (initial state), TAF produces readable tangent linear code which can be readily used.

For the adjoint code, a few interventions were made in order to deal with state variables that are required in evaluating the derivative expressions in reverse order. The time stepping loop was split into a two-level checkpointing (see e.g. [19, 8, 13]) according to `nTimeSteps` = `nOuter` ∗ `nInner`. The model state is then stored to disk once every `nOuter` time steps. At these instances the model picks up to recompute over an interval of `nInner` time steps. Over this interval only it stores the required variables at each time step to common blocks to be available in reverse mode. This enables an efficient balance of storing vs. recomputation at the cost of one additional model integration and some extra memory.

5 Coupling TLM and ADM

The model $\mathcal{M}$ implements a vector function $F : \mathbb{R}^6 \rightarrow \mathbb{R}^6$ such that $\mathbf{y} = F(\mathbf{x})$, where $\mathbf{x}$ represents the initial and $\mathbf{y}$ the final state as in Section 4. The Jacobian at some point $\mathbf{x}_0$ is denoted by $F' \in \mathbb{R}^{6\times 6}$. At each iteration of the algorithm used in Section 4 we are computing $\dot{\mathbf{y}} = F' \cdot \dot{\mathbf{x}}$ followed by $\bar{\mathbf{x}} = (F')^T \cdot \dot{\mathbf{y}}$. Instead of generating two separate tangent-linear and adjoint versions of F we are now combining both into one derivative model.

An augmented version of TLM performs the following steps for all assignments $v = f(\mathbf{u})$ in the original model: (A) Generation of code list; (B) local activity analysis and linearization; (C) optimal preaccumulation of gradient using the algorithm described in [16]; (D) computation of the inner product $\dot{v} = f' \cdot \dot{\mathbf{u}}$, where f' denotes the gradient of f; (E) storage of entries of f' on a stack (also known as the *tape*). An example is discussed in Section 6.

The local gradients are restored in reverse order to propagate adjoints backward through the code for each statement $v = f(\mathbf{u})$ as $\bar{\mathbf{u}} = \bar{v} \cdot f'$. In contrast with the method described in Section 4, these gradients are computed only once and used in both TLM and ADM. A downside of this approach is the requirement to store the entire derivative information. This may not be feasible for larger problems. On the other hand, local gradients of scalar assignments can be preaccumulated optimally [16, 15], which may lead to a further reduction of the computational effort.

Steps (A)-(D) are fully automated at the graph level and implemented as part of the XAIFBooster library that is being developed at Argonne National Laboratory. It uses a language-independent intermediate format for the semantic transformation of numerical programs called XAIF [20]. A protocol of how the transformation of the assignments in subroutine `box_timestep` was done can be found under `http://www-unix.mcs.anl.gov/~naumann/iccsa03`. The actual code generation was done manually. Work is in progress to interface the library with front-ends for both C/C++ and Fortran 95 in order to provide a new software tool for AD.

From the viewpoint of differentiation the information provided by the output of the XAIFBooster module is nearly sufficient to generate both tangent-linear and adjoint models in general. We have correct (and efficient) gradient code for all statements and the information on the corresponding locally dependent and independent variables. All we need in addition to this are correct unparsers and for adjoint models a mechanism to reverse the flow of control. The latter turned out to be trivial for the simple box model at hand. Although this approach is certainly feasible, it may not be optimal in many cases.

The full automization of the entire process is the subject of ongoing work at Argonne National Laboratory and Rice University. In collaboration with scientists at M.I.T. a new infra-structure for the implementation of AD algorithms is developed. This work is supported by NSF under its ITR program. See also `http://www.autodiff.org/ACTS` for further information on the project.

The source code of the coupled tangent-linear / adjoint model that is based on optimally preaccumulated local gradients of scalar assignments can be accessed online under `http://www-unix.mcs.anl.gov/~naumann/iccsa03`.

6 Optimal Statement-level Gradient Accumulation

Using the first assignment from subroutine `box_timestep` we sketch the algorithm used to preaccumulate the local gradients. A more complete description of the procedure as well as proofs for its optimality regarding the computational effort can be found in [16].

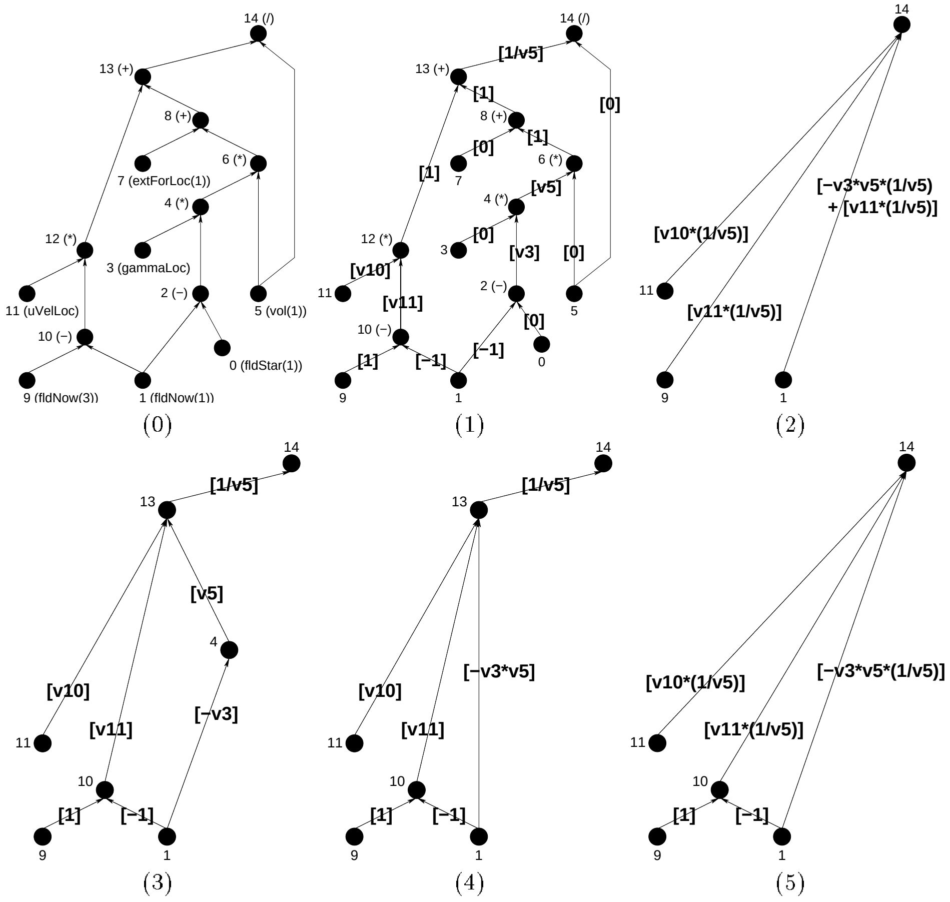

Fig. 2. Manipulation of the Computational Graph

Original statement. The original statement has the following form.

```
  dFldDt(1) = ( extForLoc(1)
&    + gammaLoc*( fldStar(1) - fldNow(1) )*vol(1)
&    + uVelLoc* (fldNow(3)-fldNow(1))) / vol(1)
```

Its computational graph **G** is shown in Figure 2(0). We are interested in the gradient of the dependent variable **dFldDt(1)** with respect to the independent variables **fldNow(1)**, **fldNow(3)** and **uVelLoc**. Our objective is convert **G** into the bipartite graph in Figure 2(2) whose edge labels represent exactly the entries of the gradient [21].

Code List. The aim of an optimized code list is to assign only those intermediate values to auxiliary variables that are used by the computation of some local partial derivative. In our case this applies only to the value of **fldNow(3)-fldNow(1)**. Thus, the number of new variables to be generated in the augmented TLM can be kept minimal. The following two assignments are generated:

```
  xb_aux_4=fldNow(3)-fldNow(1)
  dFldDt(1) =( extForLoc(1)
&        + gammaLoc*( fldStar(1) - fldNow(1) )*vol(1)
&        + uVelLoc* xb_aux_4) / vol(1)
```

Local Activity Analysis. The statement has three *active* [22] inputs (**fldNow(1)**, **fldNow(3)** and **uVelLoc**) represented by the independent vertices 1,9, and 11 in **G**. Vertices that can be reached from some independent vertex via a path in **G** are active (2,4,6,8,10,12,13,14). All other vertices are passive (0,3,5,7). Outedges of passive vertices are labeled with 0 in Figure 2(1) as the local partial derivative with respect to a passive variable vanishes identically.

Local Partial Derivatives and Elimination Procedure. The computational graph is linearized by attaching the local partial derivatives of the elemental functions with respect to their arguments to the corresponding edges. The result is shown in Figure 2(1), where the partial derivatives are enclosed in square brackets.

Constant labels of incident edges can be *folded* [23] at compile-time. If (i, j) is labeled with $c_{j,i}$ and (j, k) with $c_{k,j}$ and both $c_{j,i}$ and $c_{k,j}$ are constants then the value of $c_{j,i} \cdot c_{k,j}$ can be evaluated and (i, j) can be *front eliminated* [24] at compile-time. Notice, that (j, k) must be the only edge emanating from j since **G** is a single-expression-use graph as in [15].

Trivial edges, labelled with 1 or -1, are *back eliminated* at no extra cost. This leads to the graph in Figure 2(3). The front elimination of either $(1, 10)$ or $(9, 10)$ would not preserve optimality since 10 has more than one predecessor and its successor is not equal to the dependent vertex. Refer to [16] for proofs of these results and a more complete description of the constant folding algorithm.

The graph in Figure 2(3) is reduced to the bipartite graph in Figure 2(2) that represents the local gradient by the optimal preaccumulation algorithm for single-expression-use graphs described in [15]. Vertex 4 is eliminated (leading to the graph in Figure 2(4)) followed by 12 (Figure 2(5)) and 10. The latter does not result in any scalar floating-point multiplications since all its inedges are trivial. As a common subexpression in all gradient entries `1/v5` is assigned to an auxiliary variable. The code resulting from this elimination is shown below.

```
      c_14_13=1./vol(1)
      c_14_11=xb_aux_4*c_14_13
      c_14_10=uVelLoc*c_14_13
      c_14_1=-gammaLoc*vol(1)*c_14_13
      c_14_9=c_14_10
      c_14_1=c_14_1-c_14_10
```

Computation of Directional Derivative. Finally, the entries of the gradient are used to compute the directional derivative `dFldDt_d(1)` as

```
      dFldDt_d(1)=c_14_11*uVelLoc_d+
    &   c_14_1*fldNow_d(1)+c_14_9*fldNow_d(3)    .
```

Following this, the local gradient is stored on the tape.

Computation of Adjoints. After restoring the values for `c_14_1`, `c_14_9`, and `c_14_11` they are used to compute the corresponding adjoints as

```
      fldNow_a(3)=fldNow_a(3)+c_14_9*dFldDt_a(1)
      fldNow_a(1)=fldNow_a(1)+c_14_1*dFldDt_a(1)
      uVelLoc_a=uVelLoc_a+c_14_11*dFldDt_a(1)
      dFldDt_a(1)=0    .
```

A formal proof for the statement-level optimality of the above approach can be found in [25].

7 Comparison and Conclusion

	Separate TLM and ADM	Coupled TLM and ADM
without compiler optimization	286	265
with compiler optimization	167	183

Table 1. Runtime (in sec.) of 10,000 TLM / ADM evaluations

We compared the elapsed times for $10,000$ evaluations of the TLM / ADM. The results are displayed in Table 1. They depend strongly on the code optimizations performed by the compiler (`g77`, in this case, with or without `-O3` option activated). The coupled TLM-ADM approach generates more efficient source code and is superior if compiler optimization is switched off. On the other hand, its memory requirements are higher than those of the adjoint code generated by TAF and featuring a two-level checkpointing strategy. The repeated recomputations performed by the TAF-generated adjoint code carry not much weight for our very simple example. Moreover, the real power of the local gradient accumulation algorithms proposed in [15, 16] cannot be exploited either, because of the relative simplicity of the single statements. Enabling full compiler optimizations turns the emphasis with regard to the overall performance towards the memory traffic. There the coupled TLM-ADM looses because of the requirement to store all the local gradients of the assignments. The number of floating-point operations to be performed during the accumulation of these gradients is small.

The theoretical optimality of the local gradient accumulation routine is not sufficient to compensate the runtime increase resulting from higher memory requirements. This is not very surprising as the storage of local gradient entries increases the amount of memory required by a factor that is equal to the number of active arguments on the right hand side of the assignment compared to the strategy of storing values of intermediate variables. This is no longer the case If the preaccumulation can be applied to local Jacobians at the basic-block level this is no longer the case. There the minimization of the number of edges in the local linearized computational graph can lead to a decrease in the amount of memory required.

We conclude that the development of efficient techniques for coupling tangent-linear and adjoint models represents a challenging research area in the field of automatic differentiation. Most likely, useful approaches will represent heuristic compromises between local preaccumulation techniques and hierarchical checkpointing algorithms.

Acknowledgments

This work was supported by the National Science Foundation's Information Technology Research Program under Contract OCE-0205590 ("Adjoint Compiler Technology and Standards").

Naumann was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-ENG-38.

References

1. Wunsch, C.: What is the thermohaline circulation. Science **298** (2002) 1179
2. Farrell, B.: Optimal excitation of neutral rossby waves. J. Atmos. Sci. **45** (1988) 163–172
3. Farrell, B., Ioannou, P.: Perturbation growth and structure in uncertain flows. part i. J. Atmos. Sci. **59** (2002a) 2629
4. Tziperman, E., Ioannou, P.: Transient growth and optimal excitation of thermohaline variability. J. Phys. Oceanogr. **32** (2002) 3427
5. Stommel, H.: Thermohaline convection with two stable regimes of flow. Tellus **13** (1961) 224–230
6. Tziperman, E.: Inherently unstable climate behaviour due to weak thermohaline ocean circulation. Nature **386** (1997) 592–595
7. Griewank, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. SIAM, Philadelphia (2000)

8. Giering, R., Kaminski, T.: Recipes for adjoint code construction. ACM Transactions on Mathematical Software **24** (1998) 437–474
9. Giering, R.: Transformation of algorithms in fortran (taf). user manual version 1.3). Technical report, FastOpt (2001) http://www.fastopt.de/taf.
10. Marotzke, J., Giering, R., Zhang, K., Stammer, D., Hill, C., Lee, T.: Construction of the adjoint MIT ocean general circulation model and application to Atlantic heat transport variability. J. Geophys. Res. **104, C12** (1999) 29,529–29,547
11. Stammer, D., Wunsch, C., Giering, R., Eckert, C., Heimbach, P., Marotzke, J., Adcroft, A., Hill, C., Marshall, J.: The global ocean circulation and transports during 1992 – 1997, estimated from ocean observations and a general circulation model. J. Geophys. Res. **107(C9)** (2002a) 3118
12. Stammer, D., Wunsch, C., Giering, R., Eckert, C., Heimbach, P., Marotzke, J., Adcroft, A., Hill, C., Marshall, J.: Volume, heat and freshwater transports of the global ocean circulation 1993 –2000, estimated from a general circulation model constrained by WOCE data. J. Geophys. Res. (2002b) in press.
13. Heimbach, P., Hill, C., Giering, R.: Automatic generation of efficient adjoint code for a parallel navier-stokes solver. In J.J. Dongarra, P.M.A. Sloot and C.J.K. Tan, ed.: Computational Science – ICCS 2002. Volume 2331 of Lecture Notes in Computer Science. Springer-Verlag, Berlin (Germany) (2002a) 1019–1028
14. Heimbach, P., Hill, C., Giering, R.: An efficient exact adjoint of the parallel mit general circulation model, generated via automatic differentiation. Future Generation Computer Systems (FGCS) (2002b) submitted.
15. Naumann, U.: On optimal Jacobian accumulation for single expression use programs. Preprint ANL-MCS/P944-0402, Argonne National Laboratory (2002)
16. Naumann, U.: Automatic generation of optimal gradient code for scalar assignments. Preprint ANL-MCS/P1020-0103, Argonne National Laboratory (2003)
17. Rivin, I., Tziperman, E.: Linear versus self-sustained interdecadal thermohaline variability in a coupled box model. J. Phys. Oceanogr. **27** (1997) 1216
18. Lehoucq, R.B., Sorensen, D.C., Yang, C.: ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. SIAM, Philadelphia (1998)
19. Restrepo, J., Leaf, G., Griewank, A.: Circumvening storage limitations in variational data assimilation studies. SIAM J. Sci. Comput. **19** (1998) 1586–1605
20. Hovland, P., Naumann, U., Norris, B.: An XML-based platform for semantic transformation of numerical programs. In: M. Hamza, ed., Software Engineering and Applications, Proceedings of the Sixth IASTED International Conference. (2002) 530–538
21. Griewank, A., Reese, S.: On the calculation of Jacobian matrices by the Markovitz rule. In: [26]. (1991) 126–135
22. Hascoët, L., Naumann, U., Pascual, V.: TBR analysis in reverse-mode Automatic Differentiation. Elsevier Science (2002) under review.
23. Aho, A., Sethi, R., Ullman, J.: Compilers. Principles, Techniques, and Tools. Addison-Wesley, Reading, MA (1986)
24. Naumann, U.: Optimal accumulation of Jacobian matrices by elimination methods on the dual computational graph. Preprint ANL-MCS/P943-0402, Argonne National Laboratory (2002) To appear in Math. Prog.
25. Naumann, U.: Statement-level optimality of tangent-linear and adjoint models. Preprint ANL-MCS/P1021-0103, Argonne National Laboratory (2002)
26. Corliss, G., Griewank, A., eds.: Automatic Differentiation: Theory, Implementation, and Application. Proceedings Series, Philadelphia, SIAM (1991)